Effectiveness of Index Expressions

F.A. Grootjen and Th.P. van der Weide

University of Nijmegen, Faculty of Science, Mathematics and Computing Science,
P.O. Box 9010, 6500 GL Nijmegen, The Netherlands

Abstract. The quest for improving retrieval performance has led to the deployment of larger syntactical units than just plain words. This article presents a retrieval experiment that compares the effectiveness of two unsupervised language models which generate terms that exceed the word boundary. In particular, this article tries to show that index expressions provide, beside their navigational properties, a good way to capture the semantics of inter-word relations and by doing so, form an adequate base for information retrieval applications.

1 Introduction

The success of single-word content descriptors in document retrieval systems is both astonishing and comprehensible. Single-word descriptors are expressive, have a concise meaning and are easy to find[1]. This explains the success of word based retrieval systems. Even nowadays, modern internet search engines like Google use complicated ranking systems and provide boolean query formulation, yet are in principle still word based.

The employment of larger syntactical units than merely words for Information Retrieval purposes started in the late sixties [1], but still do not seem to yield the expected success. There are several non-trivial problems which need to be solved in order to effectively make use of multi-word descriptors:

- the introduction of multi-word descriptors boosts precision, but hurts recall.
- the manner of weighting is not obvious, especially in comparison to single-word descriptors which react suitably to standard statistically motivated weighting schemes (such as term frequency/inverse document frequency).
- it is not easy to find distant, semantically related, multi-word descriptors.

The great success of the present statistical techniques combined with such "shallow linguistic techniques" [2] has compelled the idea that deep linguistics is not worth the effort. However, advancements in natural language processing, and the ability to automatically detect related words [3, 4] justifies reevaluation.

This article attempts to compare the effectiveness of several language models capable of the unsupervised generation of multi-word descriptors. A comparison is made between standard single-word retrieval results, word n-grams and index expressions.

[1] This might be true for the English language, but for some Asian languages (for example Chinese and Vietnamese) the picture is less clear

2 Method

2.1 Measuring retrieval performance

To compare the linguistic models we use standard precision figures measured on 11 different recall values ranging from 0.0 to 1.0, and on the 3 recall values 0.2, 0.5 and 0.8. Subsequently these values are averaged over all queries.

SMART and BRIGHT The SMART system, developed by Salton [5], played a significant role in experimental Information Retrieval research. This vector space based tool offers the capability to measure and compare the effect of various weighting schemes and elementary linguistic techniques, such as stopword removal and stemming.

It became apparent that extending SMART to the specific needs of modern Information Retrieval research would be rather challenging. The lack of documentation and the style of coding complicates the extension of the system in non-trivial ways. These arguments invoked the decision to redesign this valuable system, preserving its semantic behavior, but written using modern extendible object oriented methods. The resulting system, BRIGHT, has been used in the retrieval experiments presented in this article.

Inside BRIGHT In contrast to SMART, the BRIGHT system consists of two distinguishable components: the collection specific parser and the retrieval engine. The communication between the constituents is realized by an intermediate statistical collection representation. SMART's capability to specify the input structure, and thus parameterizing the global parser, has been eliminated. Though resulting in the construction of a parser for each new collection[2], it provides the flexibility of testing elaborated linguistic techniques.

The architecture of BRIGHT is shown in Figure 1.

Test collections The principal test collection used in this article is the Cranfield test collection [1], a small standard collection of 1398 technical scientific abstracts[3]. The collection is accompanied by a rather large set of 225 queries along with human assessed relevance judgments. It consists of approximately 14,000 lines of text, and contains almost 250,000 words of which 15,000 unique.

To show that the approach presented is feasible, we tested our findings on the Associated Press newswire collection, part of the TREC dataset. This collection is approximately 800Mb big, containing 250,000 documents and 50 queries. It consists of more than 100,000,000 words of which 300,000 unique.

[2] Thanks to the object oriented structure of existing BRIGHT parsers, a parser rewrite is relatively easy.

[3] The abstracts are numbered 1 to 1400. Abstracts 471 and 995 are missing.

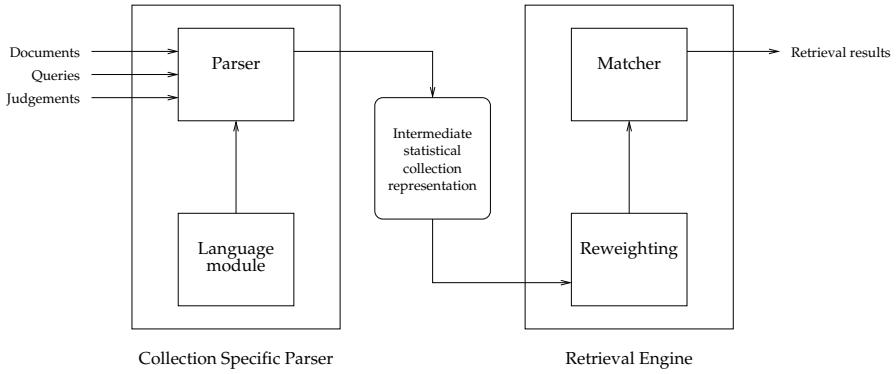

Fig. 1. The BRIGHT architecture

Baseline The retrieval results of the distinct models will be compared to the standard multiset model, without the use of a special weighting scheme (simply cosine normalization). This baseline will be referred to as `nnc` equivalent to SMART's notation for this particular weighting. The justification for not using more elaborated weighting methods is twofold:

- statistically motivated weighting schemes may mask the linguistic issues
- the purpose of the experiment is to compare different models, not to maximize (tune) retrieval results

Although one of the language models outperforms term frequency/inverse document frequency weighting (`atc`), this is of less importance regarding the scope of this article.

2.2 Beyond the word boundary

A key issue in Information Retrieval is to find an efficient and effective mechanism to automatically derive a document representation that describes its contents. The most successful approach thus far is to employ statistics of individual words, ignoring all of the structure within the document (the multiset model). Obviously indexing is not necessarily limited to words. The use of larger (syntactical) units has been the subject of research for many years. The benefit is clear: larger units allow more detailed (specific) indexes and are a way to raise precision. On the other hand, the rare occurrences of these units will hurt recall. We describe two indexing models that exceed the word boundary, namely *word n-grams* and *index expressions* and compare their retrieval performance using BRIGHT

Word n-grams The word n-gram model tries to capture inter-word relations by simply denoting the words as ordered pairs, triples etc. In effect, the n-gram model extends the multiset model with sequences of (at most n) consecutive

words in the order which they appear in the text. Consider the following document excerpt:

> An experimental study of a wing in a propeller slipstream was made in order to determine the spanwise distribution of the lift increase due to slipstream at different angles of attack of the wing and at different free stream to slipstream velocity ratios.

The 2-gram model will add, besides each word individually, the descriptor '*propeller slipstream*' which is obviously meaningful. The model is rather imprecise however, since adding the descriptor '*and at*' will probably not contribute to retrieval performance. Some researchers therefore only add n-grams consisting of non-stopwords, or consider an n-gram only worthwhile if it has a (fixed) frequency in the collection.

Index expressions As already shown before, simply using sequences of words for indexing purposes has some drawbacks:

- Sequential words are not necessarily semantically related.
- Sometimes words are semantically related, but are not sequential.

It seems plausible to look for combinations of words that are semantically related. In [3] an algorithm is presented which is capable of finding relations between words in natural language text. These relations form a hierarchical structure that is represented by index expressions.

Index expressions extend term phrases which model the relationships between terms. In this light, index expressions can be seen as an approximation of the rich concept of noun (and verb) phrases. Their philosophical basis stems from Farradane's *relational indexing* [6, 7]. Farradane projected the idea that a considerable amount of the meaning in information objects is denoted in the relationships between the terms.

Language of index expressions Let T be a set of terms and C a set of connectors. The language of index expressions is defined over the alphabet $\Sigma = T \cup C \cup \{\texttt{(}, \texttt{)}\}$ using structural induction:

(i) t is an index expression (for $t \in T$).
(ii) $e_1 \circ c\texttt{(}e_2\texttt{)}$ is an index expression (for index expressions e_1, e_2 and $c \in C$).

In this definition, the $\circ$ operator denotes string concatenation. If there are no means for confusion, we omit the parentheses when writing down index expressions.

The structural properties of these expressions provide special opportunities to support a searcher in formulating their information need in terms of a (information system dependent) query. The resulting mechanism is called *Query by Navigation* [8]. In [9] this mechanism is described from a semantical point of view. By employing the relation between terms and documents, concepts are

derived which are used as navigational pivots during Query by Navigation. Index expressions have been motivated and validated of their potential to support interactive query formulation, without assuming that the searcher is familiar with the collection. The rationale of the approach is that a searcher may not be able to formulate the information need, but is well capable of recognizing the relevance of a formulation.

Consider the following input sentence:

> The report describes an investigation into the design of minimum drag tip fins by lifting line theory.

The corresponding parsed index expression is:

```
describe SUB (report IS (the)) OBJ (investigation IS (an) INTO (design
IS (the) OF (fin IS (tip IS (drag)) IS (minimum)))) BY (theory IS (line
IS (lifting)))
```

whose structure is visualized in figure 2. Note that in this index expression, the verb-subject and verb-object relations are represented by the `SUB` and `OBJ` connectors, while apposition is represented by the `IS` connector. Using this index

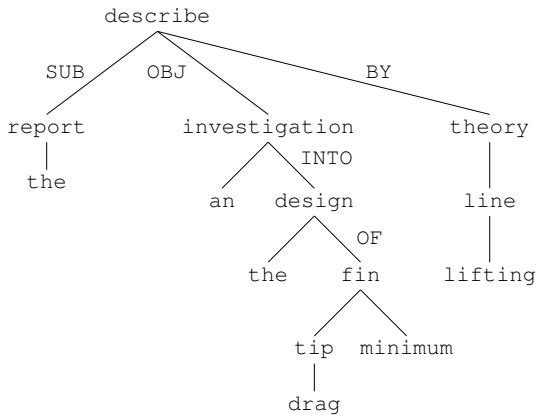

Fig. 2. Tree structure of example sentence.

expression it is possible to generate subexpressions. Simply put, subexpressions of an index expression are like subtrees of the tree structure. Preceding a more formal definition, we will introduce power index expressions, a notion similar to power sets.

Power index expressions Let $e = t \circ_{i=1}^{k} c_i e_i$ be an index expression. The set $\Lambda(e)$ of *lead expressions* belonging to e is defined as follows:

$$\Lambda(e) \stackrel{\text{def}}{\equiv} \bigcup_{(b_1,\ldots,b_k)\in\{0,1\}^k} t \circ_{i=1}^{k} (c_i \Lambda(e_i))^{b_i}$$

The power index expression belonging to e, denoted by $\mathcal{P}(e)$, is the set

$$\mathcal{P}(e) \stackrel{\text{def}}{\equiv} \Lambda(e) \cup \bigcup_{i=1}^{k} \mathcal{P}(e_i)$$

Using this definition we can now formally define what a subexpression is:

Subexpression Let e_1 and e_2 be two index expressions, then:

$$e_1 \sqsubseteq e_2 \stackrel{\text{def}}{\equiv} e_1 \in \mathcal{P}(e_2)$$

Among the subexpressions in our example sentence we find '`describe BY theory`', clearly non-sequential words having a strong semantic relation.

Instead of using all subexpressions as descriptors, we restrict ourselves to subexpressions having a maximum length.[4] In this article we evaluate the retrieval performance for 2-index, 3-index and 4-index subexpressions. Note that a similar linguistic approach which creates head-modifier frames [10] is essentially a cutdown version of index expressions, while their unnesting into head-modifier pairs generates index expressions of length 2.

3 Results

3.1 Validation results

Baseline The Cranfield baseline experiment yields the following results:

scheme	11-pt average	3-pt average
`nnc`	0.2363 (100.0%)	0.2201 (100.0%)

Word n-grams We performed retrieval runs using BRIGHT on n-grams with $1 \leq n \leq 4$ and weighting scheme `nnc`. $n = 1$ produces the multiset model (baseline). Note that, for example, the run with $n = 3$ uses word sequences of length 3 *and those smaller* as semantical units.

n	units	11-pt average	3-pt average
1	7223	0.2363 (100.0%)	0.2201 (100.0%)
2	79675	0.2554 (108.1%)	0.2401 (109.1%)
3	230870	0.2519 (106.6%)	0.2384 (108.3%)
4	422554	0.2422 (102.5%)	0.2273 (103.3%)

The results of different n-gram runs are depicted in figure 3. It is easy to see that all n-gram runs perform better than the baseline. The best improvement is obtained in the high precision - low recall area, which is not surprising, since n-grams have a more specific meaning, but occur less frequently than words. The best results are obtained for $n = 2$. As anticipated, the retrieval performance decreases slightly when n is increased, because more 'meaningless' units are generated than 'meaningful' units.

[4] The length of a index expression is the number of terms that occur in the expression.

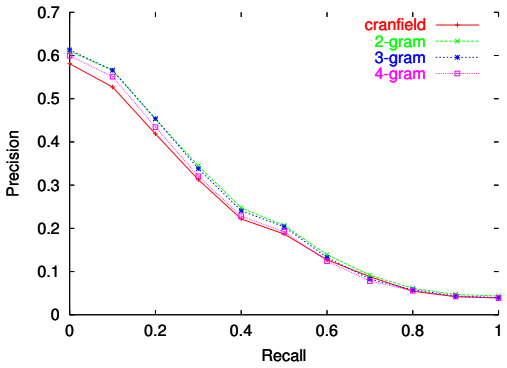

Fig. 3. n-grams compared

Index expressions As with n-grams, we use BRIGHT to measure retrieval performance for different maximum lengths of index expressions. Again, for $n = 1$ the multiset model is produced which functions as baseline. The results are presented below and visualized in figure 4.

n	terminals	11-pt average	3-pt average
1	7223	0.2363 (100.0%)	0.2201 (100.0%)
2	68061	0.2771 (117.3%)	0.2635 (119.7%)
3	206034	0.2645 (111.9%)	0.2517 (114.4%)
4	429084	0.2515 (106.4%)	0.2353 (106.9%)

The best results are obtained for $n = 2$. Obviously, long index expressions have high descriptive power, but are rare. So, similar to n-grams we notice the highest improvement in high precision - low recall area. Interesting is is that the 4-index starts off relatively good, but as soon precision drops under 0.4 it is almost indistinguishable from the baseline.

Comparing n-grams with index expressions Combining the results of the previous two sections we are capable of comparing the retrieval performance of index expressions with the performance of n-grams (see figure 5). 2-index outperforms 2-ngrams throughout the recall spectrum. The gain in performance achieved by 2-ngram is doubled by 2-index. This stresses the semantical validity of automatically generated index expressions.

3.2 TREC results

We performed two retrieval runs on the Associated Press collection: a standard word based retrieval run (baseline) and the 2-index run.

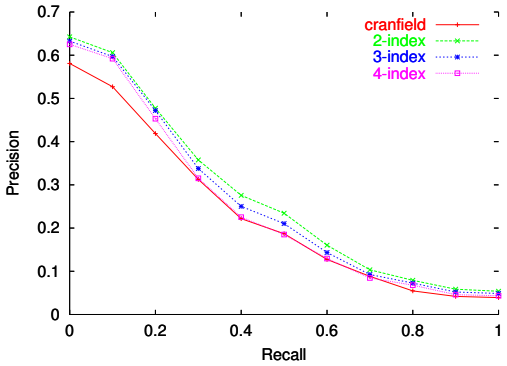

Fig. 4. index expressions

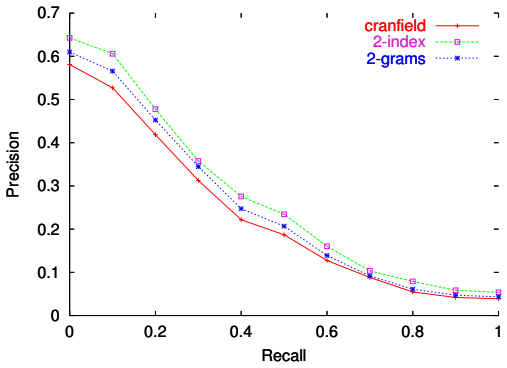

Fig. 5. index expressions vs. n-grams

type	11-pt average	3-pt average
word	0.0272 (100.0%)	0.0142 (100.0%)
2-index	0.0620 (227.9%)	0.0380 (267.6%)

The relatively low score for the baseline is primarily due to the absence of an elaborated weighting scheme. Nevertheless, the 2-index run (with the same simple weighting scheme) scores significantly better.

The resulting precision-recall data is shown in see figure 6.

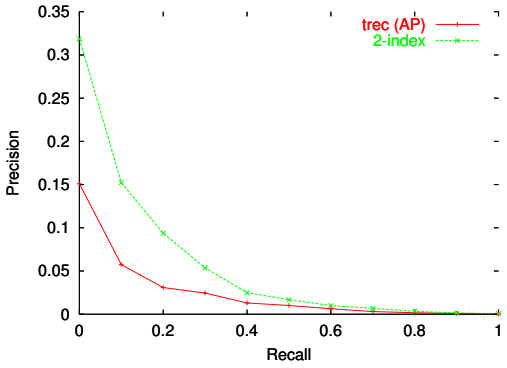

Fig. 6. Associated Press words vs. 2-index

3.3 Weighting Index expressions

In the previous experiments we treated index expressions in the same manner as terms. Because index expressions often consist of more than one word it seems reasonable to give them a higher weight than simple (single word) terms. The following experiment compares the 11-pt average retrieval performance of index expressions for several weight factors. In figure 7 we show how the retrieval performance is effected by adjusting the weight factor of index expressions having length 2. The best retrieval performance is obtained using a weight factor of approximately 2. The minimal improvement of 5% for weight factor 0 might seem strange at first glance; one might expect a gain of 0%, since eliminating index expressions with length 2 leaves us with plain terms. However, there is a mild form of stemming in the index expression model which contributes to this small gain in retrieval performance.

Studying the retrieval results of index expressions with length smaller or equal to 3, there are two changeable parameters; the weightfactor of index expressions of length 2, and the weightfactor of index expressions of length 3. This results in the 3d plot depicted in figure 8. Again, the maximal performance is obtained by doubling the weight of index expressions with length 2. For index

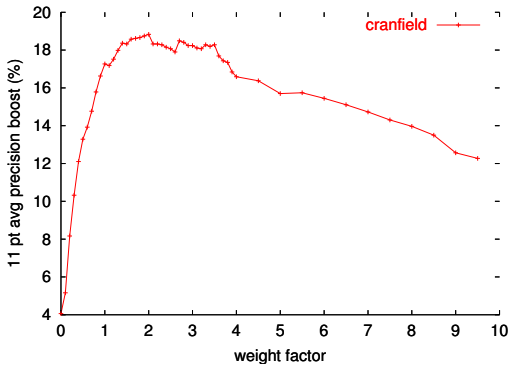

Fig. 7. Influence of weight factor

expressions with length 3 the picture is vague. Apparently the weightfactor (and the importance of these index expressions) is less obvious. According to the data, the maximal combined performance is for (2.3,3.1).

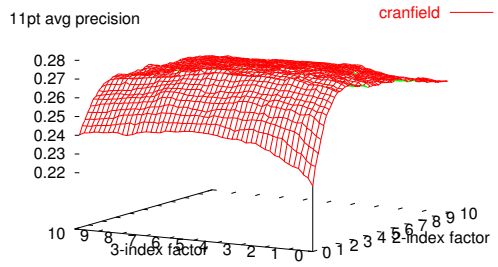

Fig. 8. influence of weight factors

4 Conclusions

As shown in this article, index expressions are suitable for capturing the semantics of inter-word relations. Experiments show that 2-indexes perform better than standard word-based retrieval runs, especially on the large TREC collection where the retrieval performance is more than doubled.

Compared to 2-grams, index expressions show an improvement of 10% on the small Canfield collection. Due to the enormous number of possible 2-grams in large collections, it was unfeasible to compare 2-grams and 2-indexes for the TREC collection.

In situations where the structure of index expressions can be exploited (as in query by navigation) they seem to form a beneficial alternative to term based systems, which is validated in [11].

References

1. Cleverdon, C.: The cranfield tests on index language devices. Aslib Proceedings (1967) 173–194
2. Sparck Jones, K.: Information retrieval: how far will *really* simple methods take you? In Hiemstra, D., de Jong, F., Netter, K., eds.: Proceedings Twente Workshop on Language Technology 14. (1998) 71–78
3. Grootjen, F.: Indexing using a grammerless parser. In: Proceedings of the 2001 IEEE International Conference on Systems, Man and Cybernetics, (NLPKE 2001), Tucson, Arizona, USA (2001)
4. Kamphuis, V., Sarbo, J.J.: Natural Language Concept Analysis. In Powers, D.M.W., ed.: Proceedings of NeMLaP3/CoNLL98: International Conference on New Methods in Language Processing and Computational Natural Language Learning, ACL (1998) 205–214
5. Salton, G., ed.: The SMART retrieval system. Prentice Hall, Englewood Cliffs (1971)
6. Farradane, J.: Relational indexing part i. Journal of Information Science **1** (1980) 267–276
7. Farradane, J.: Relational indexing part ii. Journal of Information Science **1** (1980) 313–324
8. Bruza, P., van der Weide, Th.P.: Stratified hypermedia structures for information disclosure. The Computer Journal **35** (1992) 208–220
9. Grootjen, F., van der Weide, Th.P.: Conceptual relevance feedback. In: Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics, (NLPKE 2002), Tunis (2002)
10. Koster, K.: Head/modifier frames for information retrieval. Technical report, PEKING project (2003)
11. Grootjen, F., van der Weide, Th.P.: Conceptual query expansion. Technical Report NIII-R0406, University of Nijmegen (2004)